Machine Learning, 14, 321–331 (1994)


*Extended Abstract*

# Combining Symbolic and Neural Learning

JUDE W. SHAVLIK SHAVLIK@CS.WISC.EDU
*Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, USA*

**Editor:** D. Sleeman

Date Submitted: 17 November 1992

## 1. Introduction

The last ten or so years have produced an explosion in the amount of research on machine learning. This rapid growth has occurred, largely independently, in both the symbolic and connectionist (neural network) machine learning communities. Fortunately, over the last few years these two communities have become less separate, and there has been an increasing amount of research that can be considered a hybrid of the two approaches. This extended abstract reviews some of the research that combines the symbolic and neural network approaches to artificial intelligence.

We will not attempt to define precisely the essential differences between the symbolic and connectionist approaches, as that would lead to a lengthy debate far beyond the scope of this report. If some distinction is needed, we can make the coarse approximation that symbolic approaches focus on producing discrete combinations of features, while neural approaches adjust continuous, non-linear weightings of their inputs. However, we will assume that understanding the fundamental differences between the two paradigms is a future research issue, and we will focus on some of the research that incorporates what traditionally might be considered aspects of both camps.

There are a large number of ways to combine symbolic and connectionist AI. For example, Utgoff (1988) developed an algorithm that closely integrates decision trees and perceptrons. One could also have a loosely-coupled hybrid system in which "high" level decisions are made symbolically, while "low" level ones are made by neural networks (e.g., Gallant, 1988; Pomerleau, Gowdy, & Thorpe, 1991). Recent special issues of journals (Hendler, 1989; Hinton, 1990) present additional approaches. However, rather than attempting a comprehensive review of all the symbolic/connectionist hybrid methods explored, we will focus on the framework that Figure 1 illustrates.

In this framework, the learner first *inserts* symbolic information of some sort into a neural network; it is becoming increasingly clear that a learner must make effective use of prior knowledge in order to perform well (Geman, Bienenstock, & Doursat, 1992). Once in a neural representation, it uses training examples to *refine* the initial knowledge. Finally, it *extracts* symbolic information from the trained network. The research of several groups fits nicely into this framework, and promising results have been achieved. The remainder of this paper discusses some of this research and points out open issues

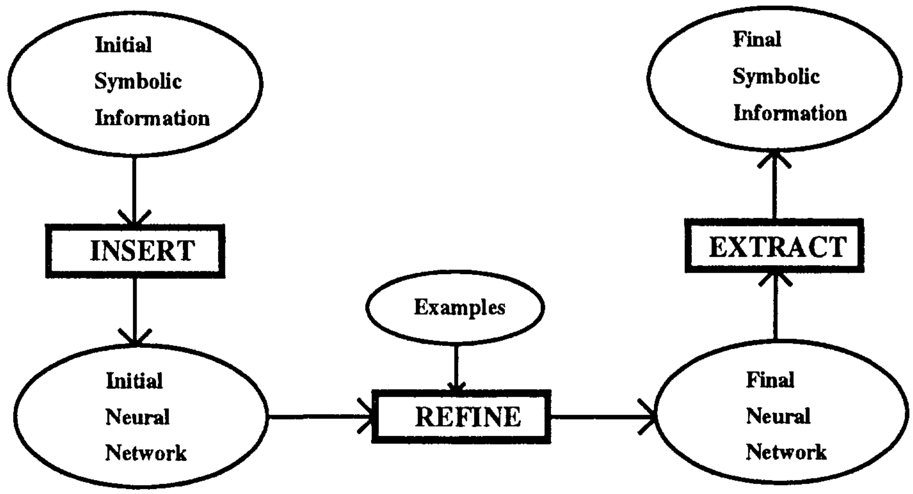


*Figure 1.* A framework for combining symbolic and neural learning

in each of the three phases. But before continuing, it should be noted that these three steps are somewhat independent and researchers have studied various combinations of them.

The remainder of this article is organized around four questions. We first consider why one should use neural networks for symbol-oriented learning tasks. We then review research that addresses questions about each of Figure 1's three phases: insertion, refinement, and extraction of symbolic information.

## 2. Why use neural networks for symbol-oriented learning tasks?

Should one avoid using connectionist methods to learn tasks that inherently deal with symbols? Are not neural networks primarily applicable to "low-level", perceptual tasks? We will argue in this section that the answer to these related questions is "no."

Over the last few years, starting with three papers published simultaneously at IJCAI-89 (Fisher & McKusick, 1989; Mooney et al., 1989; Weiss & Kapouleas, 1989) and followed by other studies (e.g., Atlas, 1990; Dietterich, Hild, & Bakiri, 1990), several groups have empirically compared symbolic learning algorithms, such as Quinlan's (1986) ID3 decision-tree algorithm, to connectionist approaches, such as Rumelhart, Hinton, and Williams's (1986) backpropagation method for training neural networks. These studies did not produce consistent results, but their coarse summary is that trained neural networks have at least comparable accuracies to induced decision trees on tasks that can be considered symbol oriented. Hence, it appears worthwhile to investigate using neural learning methods to produce and refine symbolic information.

In addition, neural network approaches have proven successful on a wide range of "real world" tasks, such as speech understanding (Lippmann, 1989), handwritten-character recognition (Le Cun et al., 1989), control of dynamic systems (Jordan & Rumelhart, 1992), gene finding (Uberbacher & Mural, 1991), and language learning (Touretzky, 1991). These experiments strongly suggest that connectionist learning is a powerful approach, and the use of neural networks with symbolic knowledge merits exploration.

Finally, it is important to note that there are connectionist architectures beyond the simple, feed-forward, single-hidden-layer neural networks. In particular, recurrent networks (Elman, 1990; Jordan, 1986), with their feedback loops and "memory", are especially appealing for application to symbolic tasks that have a sequential nature.

## 3. How can we get symbolic information into neural networks?

Assuming one is convinced of the merit of Figure 1's framework, techniques for inserting symbolic information into a neural network are needed. One can think of this preexisting information as prior knowledge about the task at hand, and the question is: how can neural networks effectively use these "hints" (Abu-Mostafa, 1990)?

One answer, the KBANN approach (Towell, Shavlik, & Noordewier, 1990; Towell, 1992), creates *knowledge-based artificial neural networks* by producing neural networks whose topological structure matches the dependency structure of the rules in an approximately-correct "domain theory" (a collection of inference rules about the current task). Figure 2 contains a simple example. KBANN has been applied to successfully refining domain theories for real-world problems such as gene finding (Towell et al., 1990), protein folding (Maclin & Shavlik, 1993), and the control of a simple chemical plant (Scott, Shavlik, & Ray, 1992)

Various groups have found that knowledge-based neural networks train faster than do "standard" neural networks (Berenji, 1991; Oliver & Schneider, 1988; Omlin & Giles, 1992; Shavlik & Towell, 1989), presumably because the initial information is used to choose a good starting point for the network. More importantly, though, experiments have shown that knowledge-based networks generalize better to future examples than do standard networks, as well as several other methods for inductive learning and theory refinement (Omlin & Giles, 1992; Maclin & Shavlik, 1993; McMillan et al., 1992; Roscheisen, Hofmann, & Tresp, 1992; Scott et al., 1992; Towell, 1992; Towell et al., 1990; Tresp, Hollatz, & Ahmad, 1993). One can attribute this improved generalization to two aspects of the insertion process. The domain theory produces a useful inductive bias by (a) focusing attention on relevant input features and (b) indicating useful intermediate conclusions (which suggest a good network topology).

Towell (1992) has shown that KBANN's knowledge-based networks better refine a domain theory than do purely symbolic theory-refinement systems. This holds even when one compares the rules extracted from the trained network to the refined rules produced by the symbolic theory-refinement systems; these results provide a justification for the complex representational shifts in Figure 1's framework. One can convert the rules that KBANN extracts to disjunctions of conjunctive rules (usually with a great increase in the number of rules), so that the two approaches are searching the same hypothesis space.

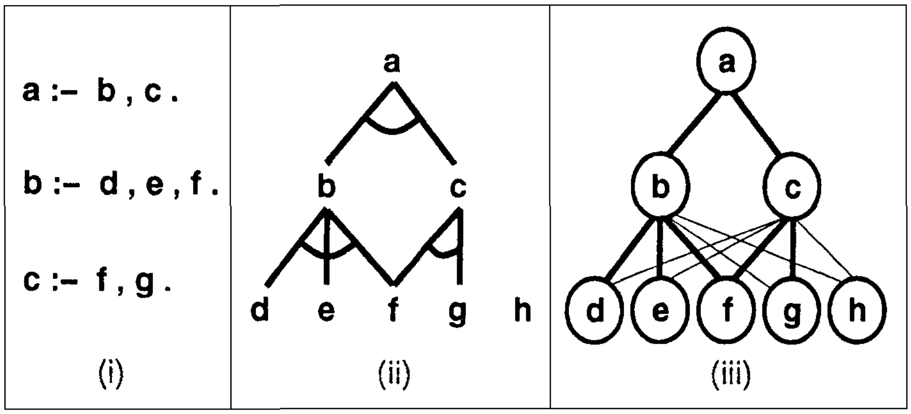


*Figure 2.* A sample application of the KBANN rule-insertion algorithm. Frame (i) contains a simple domain theory, while frame (ii) shows the dependency structure of these rules. The third frame shows the network KBANN creates. The thick lines in (iii) correspond to the dependencies in the rules; KBANN sets the weights on these links in such a manner that nodes are highly active only when the domain theory supports the corresponding deduction. Thin lines in frame (iii) represent zero-weighted links that KBANN adds to the network to allow refinement of the domain theory during neural training.

While Towell's empirical results may well be problem-specific, a broader conclusion is that searching the continuous weight space of neural networks is better on "real-world" problems than searching the combinatorial space of discrete rules – complex concepts in one representation may be much simpler in the other. A deeper understanding of the relative merits of the symbolic/connectionist and the purely symbolic approaches to theory refinement is an important open research issue.

In addition to the simple, propositional rules shown in Figure 2 and used in much of the early KBANN work, researchers have produced techniques for mapping several other forms of prior knowledge into networks. Fu (1989) and Mahoney and Mooney (1993) map rules containing certainty factors. Berenji (1991) and Masuoka et al.(1990) map fuzzy-logic rules, while McMillan, Mozer, and Smolensky (1992) use gating networks (Jacobs et al., 1991) to map production rules. Scott et al.(1992) and Roscheisen et al. (1992) map mathematical equations, demonstrating that the KBANN approach does not require logic-oriented domain theories. Finally, several groups have mapped (generalized) finite-state grammars into recurrent neural networks (Fransconi et al., 1991; Maclin & Shavlik, 1993; Omlin & Giles, 1992; Scott et al., 1992). Generalized finite-state grammars are particularly interesting to the theory-refinement community, as one can view them as state-dependent domain theories, a richer type of domain theory than is usually studied in this subfield of machine learning. These approaches differ from KBANN to various degrees, but the essential idea is the same: use prior knowledge to decide how to initialize a neural network.

There are several open questions regarding the knowledge-insertion process. We would like to know what other types of prior knowledge can be inserted into networks. For example, methods are lacking for inserting first-order theories. The last few years have seen much progress in inductive logic programming (Muggleton, 1992; Quinlan, 1990), and it would be useful to see if (and how well) neural networks can refine rules containing variables. To do so, one needs to devise methods for dealing with unbounded symbolic structures in neural networks (whose size is usually fixed following training). Recurrent networks provide one method of dealing with unbounded structures, and Pollack's (1990) recursive auto-associative memories provide another. Also relevant is research on teaching networks to recognize context-free grammars by having them learn how to use a stack (Das et al., 1993; Giles et al., 1990; Mozer & Das, 1993). Unbounded structures, such as stacks, can be handled in fixed-size networks by somehow altering resolution (in some sense) so that the product of the information being stored and its resolution equals a constant.

Towell (1992) has shown that knowledge-based networks are good at deleting irrelevant information in approximately-correct domain theories, but do not handle "impoverished" domain theories as well. Hence, another open issue is how to deal with domain theories that are incomplete. We will return to this topic in the next section.

Converting symbolic information to a neural-network representation, followed by connectionist learning, has been shown useful by several research groups. This leaves us with the central question about the insertion phase:

> How can we re-represent symbolic knowledge and learning tasks so that powerful numeric-optimization search methods are applicable?

## 4. How can network refinement be guided by symbolic knowledge?

Once prior knowledge is inserted into a network, it has to be refined and enhanced. A simple way of doing this is to run backpropagation, or some other standard connectionist training procedure, on the training examples. However, there are two ways to use symbolic information to improve training: (1) one could use symbolic learning methods and ideas to focus the adjustment of the network, both its weights and topology, and (2) one might alter backpropagation to better match the symbolic nature of a given problem. In this section we discuss both of these approaches.

One might ask, where is the symbolic learning in the approaches presented so far? One answer is that domain-theory refinement, which is what knowledge-based networks do, addresses the incorrect-theory problem of explanation-based learning. In fact, this perspective was the initial motivation for the KBANN research (Shavlik & Towell, 1989). (Recently, Mitchell and Thrun (1993) proposed an explanation-based, though non-symbolic, method for training neural networks for reinforcement learning tasks.)

But what about performing symbolic inductive learning in conjunction with neural learning? As mentioned above, Utgoff's (1988) perceptron trees are one method for doing so, but his algorithm is not applicable to the refinement of prior knowledge. Recall that in knowledge-based networks, input features fall into two classes: those that are mentioned in the domain theory and those that are not. Since the domain theory can be imperfect, one cannot ignore the unmentioned input features; they are typically connected to other units with low-weighted links. Towell and Shavlik (1992) proposed a technique that uses symbolic inductive learning to identify good input features, which are then weighted more heavily. They found this preprocessing of the network led to better generalization.

Also, as mentioned, a domain theory may be missing a number of rules. Hence, it will be mapped into a network that is too small. In order to learn these missing rules, additional nodes will have to be added to the network during training. Opitz and Shavlik (1993) developed an algorithm that interprets networks symbolically to decide where to add new nodes.

There have been several changes to standard connectionist learning motivated by symbolic problems. Rather than minimizing mean-squared error, the cross-entropy error function (Hinton, 1989) is a better choice for knowledge-based networks (see Towell (1992) for an explanation). Refining rules with certainty factors requires the use of a different activation function for nodes (Fu, 1989; Mahoney & Mooney, 1993). One may wish to constrain weight changes to maintain the symbolic interpretation of the network (McMillan et al., 1992). Finally, networks often decay their weights toward zero during training (Hinton, 1986). Weights in knowledge-based networks should decay toward their initial values (Hinton, personal communication; Tresp et al., 1993), thereby encouraging the network to preserve the knowledge in the initial domain theory.

There are several open questions regarding the use of symbolic information to aid the refinement step. How can one detect that extra nodes are needed to generalize well, and where are the best places to add them? Folk wisdom says that backpropagation does not work well in networks with many layers of hidden units, because the error signal becomes too diffuse. Can one use symbolic information to focus the back-propagated error signal, especially in deep networks? Deep networks often occur when basing the network topology on the dependency structure of a rule base, so this problem is exacerbated in knowledge-based networks. Finally, do we need to prevent distributed representations (Hinton, 1986) from evolving during training? Since hidden units in knowledge-based networks initially have a symbolic meaning, it seems that distributed representations are undesirable; however, there could be some way to take advantage of distributed representations.

In summary, the central question about the refinement phase is:

How can symbolic knowledge about the task at hand guide network refinement?

## 5. How can we extract symbolic knowledge from trained neural networks?

The third phase of Figure 1's framework involves extracting symbolic information (e.g. rules) from a trained network, which need not originally be knowledge-based. Why is this important? Rule extraction can help one understand what the "black box" network has learned. If the network produced a scientifically-interesting discovery, it would be nice if this were made explicit. Also, one may wish that a trained system would produce explanations of its future decisions. Finally, one may want to manipulate the results of learning in another system, such as a planner.

Several people have developed methods for extracting rules from standard networks. Gallant (1988), Saito and Nakano (1988), and Fu (1991) proposed algorithms that consider various ways that a node's weighted input can exceed its threshold, and convert each of these situations into a rule. However, these approaches can require an exponential number of rules (in terms of the number of network weights) to re-represent a node. Towell and Shavlik (1993) developed a method that produces about one "N out of M" rule for each node. They found their algorithm extracted comprehensible rules while maintaining the accuracy of the trained network. However, their approach only works well on knowledge-based networks, as it requires that weights cluster into a few groups; the "soft-weight sharing" technique of Nowlan and Hinton (1992) can improve the performance of Towell and Shavlik's algorithm on standard networks (Craven & Shavlik, 1993). Finally, McMillan et al. (1992) simply project trained nodes to the closest valid rule, while Hayashi (1991) extracts a small number of fuzzy-logic rules from a trained network.

The above methods analyze the weights going into nodes. Cleermans, Servan-Schreiber, and McClelland (1989) and Giles et al. (1992) have a different perspective. They investigate extracting finite-state automata from recurrent networks, and their methods focus on the activation patterns of the hidden units. Their approaches assume that these patterns represent some sort of internal state. The extraction algorithms cluster these patterns and view each cluster as a state in an automaton. The next step runs the training examples through the trained network to obtain the state transitions, after which traditional algorithms minimize the automaton.

A major question with rule extraction is: how does one measure comprehensibility? An extraction algorithm must produce reasonably comprehensible rules, but without a good measure it is hard to compare alternative approaches. A second open issue relates to the refinement phase: how should this task be altered in support of rule extraction? Possibly the network can be constrained to always lie in the "comprehensible" portion of weight space, whatever that might be. Related to this, the hidden units in knowledge-based networks generally have symbolic names attached to them, and if one is going to use these labels for the extracted rules, one needs to ensure that the symbol-node correspondence is not altered during training. This is one reason why the formation of distributed representations during training can be harmful. Finally, conceptually clustering hidden-unit activations is a promising research area for symbolic machine learning. Finding and describing clusters can provide insight into the distinctions made by the network. For example, Sejnowski and Rosenberg (1987) manually analyzed clusters developed on the NETtalk task, with some success.

To wrap up this section, the central question about rule extraction is:

> How can we extract a small and comprehensible symbolic version of a trained network without losing (much) accuracy?

## 6. Conclusion

Connectionist machine learning has proven to be a fruitful approach, and it makes sense to investigate systems that combine the strengths of the symbolic and connectionist approaches to AI. Over the past few years, researchers have successfully developed a number of such systems. This article summarizes one view of this endeavor, a framework that encompasses the approaches of several different research groups. This framework (see Figure 1) views the combination of symbolic and neural learning as a three-stage process: (1) the insertion of symbolic information into a neural network, thereby (partially) determining the topology and initial weight settings of a network, (2) the refinement of this network using a numeric optimization method such as backpropagation, possibly under the guidance of symbolic knowledge, and (3) the extraction of symbolic rules that accurately represent the knowledge contained in a trained network. These three components form an appealing, complete picture—approximately-correct symbolic information in, more-accurate symbolic information out—however, these three stages can be independently studied. In conclusion, the research summarized in this paper demonstrates that combining symbolic and connectionist methods is a promising approach to machine learning.

## Acknowledgements

I wish to thank Geoff Towell, Mick Noordewier, Rich Maclin, Gary Scott, Mark Craven, Dave Opitz, Derek Zahn, Charlie Squires, and Kevin Cherkauer - all members of the University of Wisconsin Machine Learning Research Group at one time or another and major contributors to the ideas presented in this paper. Discussions with Ray Mooney, Tom Dietterich, and Geoff Hinton also substantially influenced this discussion. This work was partially supported by Office of Naval Research Grants N00014-90-J-1941 and N00014-93-1-0998, National Science Foundation Grant IRI-9002413, and Department of Energy Grant DE-FG02-91ER61129.

## References

Abu-Mostafa, Y. S. (1990). Learning from hints in neural networks. *Journal of Complexity, 6*, 192-198.

Atlas, L., Cole, R., Connor, J., El-Sharkawi, M., Marks II, R., Muthusamy, Y., & Barnard, E. (1990). Performance comparisons between backpropagation networks and classification trees on three real-world applications. In *Advances in Neural Information Processing Systems (Vol. 2)*, D. Touretzky (ed.), San Mateo, CA: Morgan Kaufmann.

Berenji, H. R. (1991). Refinement of approximate reasoning-based controllers by reinforcement learning. *Proceedings of the Eighth International Machine Learning Workshop* (pp. 475-479), Evanston, IL: Morgan Kaufmann.

Cleermans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation, 1*, 372-381.

Craven, M.W. & Shavlik, J.W. (1993). Learning symbolic rules using artificial neural networks. *Proceedings of the Tenth International Machine Leraning Conference*, pp. 73-80, Amherst, MA.

Das, S., Giles, C. L., & G. Z. Sun (1993). Using hints to successfully learn context-free grammars with a neural network pushdown automaton. In *Advances in Neural Information Processing Systems (Vol. 5)*, S. Hanson, J. Cowans, & L. Giles (eds.), San Mateo, CA: Morgan Kaufmann.

Dietterich, T. G., Hild, H., & Bakiri, G. (1990). A comparative study of ID3 and backpropagation for English text-to-speech mapping. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 24-31), Austin, TX: Morgan Kaufmann.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*, 179-211.

Fisher, D. H. & McKusick, K. B. (1989). An empirical comparison of ID3 and back-propagation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 788-793). Detroit: Morgan Kaufmann.

Frasconi, P., Gori, M., Maggini, M., & Soda, G. (1991). A unified approach for integrating explicit knowledge and learning by example in recurrent networks. *Proceedings of the International Joint Conference on Neural Networks*, (pp. 811-816). Seattle: IEEE Press.

Fu, L. M. (1989). Integration of neural heuristics into knowledge-based inference. *Connection Science, 1*, 325-340.

Fu, L. M. (1991). Rule learning by searching on adapted nets. *Proceedings of the Ninth National Conference on Artificial Intelligence* (pp. 590-595). Anaheim, CA: AAAI Press.

Gallant, S. I. (1988). Connectionist expert systems. *Communications of the ACM, 31*, 152-169.

Gemen, S., Bienenstock, E. & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation, 4*, 1-58.

Giles, C., Sun, G., Chen, H., Lee, Y., & Chen, D. (1990). Higher order recurrent networks and grammatical inference. In *Advances in Neural Information Processing Systems (Vol. 2)*, D. Touretzky (ed.), San Mateo, CA: Morgan Kaufmann.

Giles, C., Miller, C., Chen, D., Chen, H., Sun, G., & Lee, Y. (1992). Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation, 4*, 393-405.

Hayashi, Y. (1991). A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosis. In *Advances in Neural Information Processing Systems (Vol. 3)*, R. Lippmann, J. Moody, & D. Touretzky (eds.), San Mateo, CA: Morgan Kaufmann.

Hendler, J. A. (ed.) (1989). Special issue on hybrid systems (symbolic/connectionist). *Connection Science, 1*.

Hinton, G. E. (1986). Learning distributed representations of concepts. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (pp. 1-12). Amherst, MA: Lawrence Erlbaum.

Hinton, G. E. (1989). Connectionist learning procedures *Artificial Intelligence, 40*, 185-234.

Hinton, G. E. (ed.) (1990). Special issue on connectionist symbol processing. *Artificial Intelligence, 46*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*, 79-87.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, (pp. 531-546), Amherst, MA: Lawrence Erlbaum.

Jordan, M. I. & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science, 16*, 307-354.

Le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*, 541-551.

Lippmann, R. P. (1989). Review of neural networks for speech recognition. *Neural Computation, 1*, 1-38.

Maclin, R. & Shavlik, J. W. (1993). Using knowledge-based neural networks to improve algorithms: Refining the Chou-Fasman algorithm for protein folding. *Machine Learning*, 11, 195-215.

Mahoney, J. J. & Mooney, R. J. (1993). Combining neural and symbolic learning to revise probabilistic rule bases. In *Advances in Neural Information Processing Systems (Vol. 5)*, S. Hanson, J. Cowans, & L. Giles (eds.), San Mateo, CA: Morgan Kaufmann.

Masuoka, R., Watanabe, N., Kawamura, A., Owada, Y., & Asakawa, K. (1990). Neurofuzzy system - fuzzy inference using a structured neural network. *Proceedings of the International Conference on Fuzzy Logic & Neural Networks* (pp. 173-177), Iizuka, Japan.

McMillan, C., Mozer, M. C., & Smolensky, P. (1992). Rule induction through integrated symbolic and subsymbolic processing. In *Advances in Neural Information Processing Systems (Vol. 4)*, J. Moody, S. Hanson, & R. Lippmann (eds.), San Mateo, CA: Morgan Kaufmann.

Mitchell, T. M. & Thrun, S. (1993). Explanation-based neural network learning for robot control. In *Advances in Neural Information Processing Systems (Vol. 5)*, S. Hanson, J. Cowans, & L. Giles (eds.), San Mateo, CA: Morgan Kaufmann.

Mooney, R., Shavlik, J., Towell, G., & Gove, A. (1989). An experimental comparison of symbolic and connectionist learning algorithms. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 775-780). Detroit: Morgan Kaufmann. (An extended version appeared in *Machine Learning, 6*, 111-143, 1991.)

Mozer, M. C. & Das, S. (1993). A connectionist chunker that induces the structure of context-free languages. In *Advances in Neural Information Processing Systems (Vol. 5)*, S. Hanson, J. Cowans, & L. Giles (eds.), San Mateo, CA: Morgan Kaufmann.

Muggleton, S. (1992). *Inductive Logic Programming*, London: Academic Press.

Nowlan, S. J. & Hinton, G. E. (1992). Simplifying neural networks by soft weight-sharing. In *Advances in Neural Information Processing Systems (Vol. 4)*, J. Moody, S. Hanson, & R. Lippmann (eds.), San Mateo, CA: Morgan Kaufmann.

Oliver, W. L. & Schneider, W. (1988). Using rules and task division to augment connectionist learning. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society* (pp. 55-61), Montreal: Lawrence Erlbaum.

Omlin, C. W. & Giles, C. L. (1992). Training second-order recurrent neural networks using hints. *Proceedings of the Ninth International Conference on Machine Learning* (pp. 361-366), Aberdeen, Scotland: Morgan Kaufmann.

Opitz, D. & Shavlik, J.W. (1993). *Heuristically expanding knowledge-based neural networks. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 1360- 1365), Chamborg, France: Morgan Kaufmann.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence, 46*, 77-105.

Pomerleau, D.A., Gowdy, J., & Thorpe, C.E. (1991). Combining artificial neural networks and symbolic processing for autonomous robot guidance. *Engineering Applications of Artificial Intelligence, 4*, 279-285.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81-106.

Quinlan, J. R. (1990). Learning logical definitions from relations, *Machine Learning, 5*, 239-266.

Roscheisen, M., Hofmann, R. & Tresp, V. (1992). Neural control for rolling mills: Incorporating domain theories to overcome data deficiency. In *Advances in Neural Information Processing Systems (Vol. 4)*, J. Moody, S. Hanson, & R. Lippmann (eds.), San Mateo, CA: Morgan Kaufmann.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing (Vol. 1)*, D. E. Rumelhart & J. L. McClelland (eds.). Cambridge, MA: MIT Press.

Saito, K. & Nakano, R. (1988). Medical diagnostic expert system based on the PDP model. *Proceedings of the IEEE International Conference on Neural Networks* (pp. 255-262). IEEE Press.

Scott, G., Shavlik, J., & Ray, W. (1992). Refining PID controllers using neural networks. *Neural Computation, 4*, 746-757.

Sejnowski, T. J. & Rosenberg, C. (1987). Parallel networks that learn to pronounce English text. *Complex Systems, 1*, 145-168.

Shavlik, J. W. & Towell, G. G. (1989). An approach to combining explanation-based and neural learning algorithms. *Connection Science, 1*, 233-255.

Touretzky, D. S. (ed.) (1991). Special issue on connectionist approaches to language learning. *Machine Learning, 7*.

Towell, G. G. (1992). *Symbolic Knowledge and Neural Networks: Insertion, Refinement, and Extraction.* Doctoral dissertation, Madison, WI: University of Wisconsin, Computer Sciences Department.

Towell, G. G., Shavlik, J. W. & Noordewier, M. O. (1990). Refinement of approximately correct domain theories by knowledge-based neural networks. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 861-866), Boston: AAAI Press.

Towell G. G. & Shavlik, J. W. (1992). Using symbolic inductive learning to improve knowledge-based neural networks. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 177-182), San Jose, CA: AAAI Press.

Towell, G. G. & Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13, 71-101.

Tresp, V., Hollatz, J., & Ahmad, S. (1993). Network structuring and training using rule-based knowledge. In *Advances in Neural Information Processing Systems (Vol. 5)*, S. Hanson, J. Cowans, & L. Giles (eds.), San Mateo, CA: Morgan Kaufmann.

Uberbacher, E. C. & Mural, R. J. (1991). Locating protein coding regions in human DNA sequences by a multiple sensor - neural network approach. *Proceedings of the National Academy of Sciences, 88*, 11,261-11,265.

Utgoff, P. E. (1988). Perceptron trees: A case study in hybrid concept representations. *Proceedings of the Seventh National Conference on Artificial Intelligence* (pp. 601-606). St. Paul, MN: Morgan Kaufmann.

Weiss, S. & Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 781-786). Detroit: Morgan Kaufmann.